Online Learning for Data Streams With Incomplete Features and Labels

Dianlong You, *Member, IEEE*, Huigui Yan, Jiawei Xiao, Zhen Chen, Di Wu, *Member, IEEE*, Limin Shen, *Member, IEEE*, and Xindong Wu, *Fellow, IEEE*

***Abstract*—Online learning is critical for handling complex data streams in Big Data-related applications. This study explores a new online learning problem where both the features and labels are incomplete. Such incompleteness poses a critical challenge in determining the latent relationship between incomplete features and labels. Unfortunately, existing online learning methods only consider a few cases of incomplete feature spaces, such as trapezoidal, evolvable, and capricious data streams, limiting their applicability to this problem. To bridge this gap, this study proposes a novel algorithm of Online Learning for Data Streams with Incomplete Features and Labels (OLIFL). OLIFL imposes no constraints on changing patterns of feature space and does not require all instances to be labeled with two-fold ideas. First, OLIFL explores the informativeness of individual features to update the classifier by dynamically maintaining global feature space and updating the informativeness matrix. Second, it estimates the label confidence of unlabeled instances to control their negative effects by limiting the error upper bound. Extensive experiments on benchmark datasets are conducted in five scenarios: three incomplete feature (trapezoidal, evolvable, and capricious) spaces, and two incomplete labels (only missing labels and missing both features and labels). In addition, we explore the sensitivity of the model to parameters, and its usability and response efficiency in handling concept drifts. The results show that OLIFL significantly outperforms its rivals. Moreover, we use OLIFL to classify a movie review task as real application verification.**

***Index Terms*—Online learning, data stream, incomplete feature space, incomplete labels.**

Manuscript received 23 October 2022; revised 1 September 2023; accepted 27 February 2024. Date of publication 20 March 2024; date of current version 7 August 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62276226, Grant 62176070, Grant 62102348, and Grant 62120106008, in part by the Natural Science Foundation of Hebei Province under Grant F2021203038, and Grant F2022203012, and in part by the S&T Program of Hebei under Grant 236Z7725G and Grant 236Z0103G. Recommended for acceptance by Yanyan Shen. *(Corresponding author: Di Wu.)*

Dianlong You, Huigui Yan, Jiawei Xiao, Zhen Chen, and Limin Shen are with the School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China (e-mail: youdianlong@sina.com; rocky_yhg@163.com; xiaojiaweix@163.com; zhenchen@ysu.edu.cn; shenllmm@sina.com).

Di Wu is with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: wudi.cigit@gmail.com).

Xindong Wu is with the Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Ministry of Education, Hefei 230009, China (e-mail: xwu@hfut.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/TKDE.2024.3374357, provided by the authors.

Digital Object Identifier 10.1109/TKDE.2024.3374357

I. Introduction

In THIS era of Big Data, data are automatically and continuously generated in many industrial applications, such as real-time ecological monitoring [1], constantly updated hot topics on Microblog and Twitter [2], [3], real-time filtering of webspam [4], [5], online recommendation systems [6], and stock exchange market [7]. These data streams are continuous and arrive sequentially, necessitating rapid data analysis. Online learning is an effective and efficient approach to processing data stream [8], [9]. It extracts the latent knowledge and patterns from data streams to learn a model in real time. Traditional online learning methods impose a fixed feature space constraint on the data streams [10]. However, in many real applications, data streams commonly have a dynamic feature space. Moreover, dynamic features usually have numerous missing data due to various uncontrollable factors, such as instrument failures, privacy/security policies, and human errors. Therefore, dynamic and incomplete feature (DIF) space is a fundamental and crucial characteristic of data streams in online learning.

Recently, some online learning methods have been proposed to handle the DIF space. Representative examples are online learning with streaming features (OLSF) [11] and feature evolvable streaming learning (FESL) [12]. However, they have special constraints on the DIF space. OLSF is designed for trapezoidal data streams, i.e., subsequent instances must include more features than preceding instances. FESL is designed for feature evolvable data streams, i.e., several subsequent consecutive instances must include all features of preceding instances. Undoubtedly, both OLSF and FESL are impractical because the DIF space is typically capricious in real scenarios.

To handle the capricious DIF space, online learning from varying features (OLVF) [13] and generative learning with streaming capricious (GLSC) [14] have been proposed. They pose no constraint on the DIF space pattern. However, they require that labels of instances be available. In real applications, data streams usually appear at high speeds. Labeling all instances as soon as they arrive manually is impractical. In other words, many instances of data streams may have missing labels during online learning; Directly ignoring these unlabeled instances is unreasonable because they also contain valuable information. Semi-supervised learning (SSL) is a successful and popular approach for addressing missing labels. Up till date, although some studies have adopted SSL to handle data streams with

missing labels [15], [16], [17], [18], such as online reliable semi-supervised learning (REAL) [19], they only consider the fixed and complete feature space, therefore, they cannot handle the data streams with DIF space.

In this study, we explore a new online learning problem wherein the features and labels of data streams are incomplete, i.e., Online Learning for Data Streams with Incomplete Features and Labels (OLIFL). The crux of OLIFL lies in how to establish the latent relationships between DIF space and incomplete labels, which raises the following three challenges:

1) Incomplete feature space contains some uncertain patterns, making it difficult for the model to learn data stream distributions.
2) Incomplete labels cannot be directly used to train models during online learning. Effectively addressing unlabeled instances is the key to improving model performance.
3) DIF space inevitably leads to distribution changes [20] and concept drifts, making the model trained before incapable of representing the data distributions at present [21], which affects the usability of the mode.

To address these challenges, we propose a novel OLIFL algorithm with a two-fold idea: 1) it explores the informativeness of individual features to update the classifier by dynamically maintaining global feature space along with the informativeness matrix, and 2) it estimates confidence of pseudo labels for unlabeled instances to control their negative effects by limiting the error upper bound. To evaluate the OLIFL, we compare it with related online learning algorithms on benchmark datasets in three kinds of incomplete feature space (trapezoidal, evolvable, and capricious), and two situations of incomplete labels (incomplete labels only and incomplete both features and labels). Additionally, we evaluate the usability and response efficiency of OLIFL on different missing features/labels and concept drifts. The results demonstrate that OLIFL significantly outperforms its competitors. The contributions of this study are summarized as follows:

1) This is the first study to explore the problem of online learning from data streams with incomplete features and labels, which is more commonly encountered in real applications.
2) We design a novel online learning approach(OLIFL) that imposes no constraints on the changing patterns of feature space and does not require all instances to be labeled. Experiments on both real and synthetic datasets show that OLIFL is significantly more generalized and robust to different streaming data patterns than existing approaches.
3) To promote reproducible research, we open-access our resource code implementations at the following link: https://github.com/youdianlong/OLIFL.git.

II. Related Work

This section discusses studies related to ours from two perspectives: **Online learning** (OL) and **Semi-supervised Learning** (SSL). As an important branch of machine learning algorithms, OL can identify instances promptly from continuously arriving data streams [22]. Furthermore, several SSL algorithms for streaming and static data have been proposed as a natural way to handle missing labels. Thus, we review the following three related topics: 1) Online learning for data streams with incomplete features, 2) Online learning for data streams with incomplete labels, 3) Semi-supervised Learning, and 4) Concept drift.

A. Online Learning for Data Streams With Incomplete Features

Online learning from DIF space explores and exploits arbitrary varying feature spaces to train classifiers. Related studies are as follows: OLSF [11] learns only from trapezoidal data streams with a monotonical increase in the feature space based on passive–aggressive update criteria. It constructs the classifier by truncating a fraction of the lowest weights after projecting the classifier to its l_1 ball at each step. Furthermore, OLSF learns only from trapezoidal data streams in which both instances and features increase monotonically. FESL [12] can learn a linear projection matrix to address evolving feature spaces with small overlaps between old and new features. Note that the projection cannot provide exact information when the relationship between the old and new feature spaces is nonlinear. Furthermore, learning the projection and training multiple classifiers may incur significant computational costs in high-dimensional data. Moreover, FESL learns only from the feature evolvable data streams wherein feature spaces evolve in batches. OLVF [13] deals with varying feature spaces by constructing two classifiers for feature spaces and instances. The instance classifier is constantly updated as arriving instances are continuously learned, whereas the feature space classifier uses projection confidence to estimate the probability of correct prediction by the instance classifier. When the incompleteness of features is significant, the feature space classifier may dramatically degrade the performance of the instance classifier. GLSC [14] completes unobserved features by constructing a generative graphical model to handle data streams with capricious feature spaces. The model can linearly indicate the correlation between features using a latent matrix. Furthermore, the computational cost of maintaining a latent matrix is extremely high.

B. Online Learning for Data Streams With Incomplete Labels

Online learning from data streams with incomplete labels explores learning from data streams with randomly missing labels and whose structure does not make any assumptions. The main studies include as following. *Semi-supervised learning on data streams via temporal label propagation* (TLP) [18] utilize a graph-based label propagation to infer the labels of instances. However, TLP does not learn from data streams with incomplete features. Meanwhile, TLP is not strictly online processing because it relies on recent instances to update the graph. *Online reliable semi-supervised learning on evolving data streams(REAL)* [23] utilizes micro-clusters to learn initial data, then predicts the labels of receiving instances using the k-NN classifier, and exploits these instances to update the classification model. However, REAL requires labeled instances to initialize the classifier, which is inapplicable for randomly missing labeled data streams, and incomplete stream data. *An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams(SPASC)* [24] can

classify non-stationary data streams using a pool of classifiers. SPASC learns a batch of labeled instances and then explores the labeled batch to update the classifiers in the pool. However, SPASC ignores missing features and only identifies recurring concept drifts instead of stochastic drifts. Zhu et al. [25] proposed an incremental/decremental max-flow algorithm for online semi-supervised learning from data streams. The algorithm can dynamically update a graph learned from labeled and unlabeled data by computing the minimum slice of the current pass-through Max-flow and assigning the minimum number of similar instance pairs to different categories to accommodate the changes of new and old instances. However, the method consumes a significant amount of time and ignores concept drifts while updating the graph.

C. Semi-Supervised Learning

Classical SSL models adopted for data stream scenarios are as follows. *Generative models* are early semi-supervised algorithms that optimize the fitness of the model by propagating label information from labeled to unlabeled data using some generative models like Gaussian mixture models [26]. To handle streaming data, the model in [27] uses only labeled data to train a set of classifiers. However, it consumes a significant amount of time and space because of multiple integrated classifiers. *Graph-based models* [28] and [29] propagate label information over the inherent structure of data based on the manifold assumption, which is revealed by both unlabeled and labeled data. Several techniques in [25] and [18] are considered graph-based approaches. A graph is constructed from labeled and unlabeled instances, which are dynamically updated by adding and removing obsolete data. Due to the constraint of manifold assumption, these algorithms perform poorly on sophisticated data streams. *Self-training models* are initially trained with only labeled data, and then unlabeled instances with high confidence and their predicted labels to extend the training set and update the model. In the data stream scenarios, the model in [30] uses a self-training approach to create an ensemble classifier, resulting in a high time penalty. *Co-training models* [31], [32] train two or more classifiers using feature subsets, and then these classifiers extend the labeled sets of other classifiers by predicting unlabeled data. Similarly, [33] proposes the SCoforest algorithm for streaming data via chunk-by-chunk co-training and random forest. Because of the integration of multiple classifiers and block learning of instances, the models consume a significant amount of time and space. Additionally, some classical methods such as *Label propagation through linear neighbourhoods*(LNP) [34] does not apply to data streams. LNP constructs a graph using the neighbourhood information and propagates the labels of labeled data to unlabeled data on the graph using a method similar to label propagation.

D. Concept Drift

Concept drift can be generally categorized according to the speed of change of the distribution [35], e.g., sudden drifts and gradual drifts. Most algorithms use detectors [36] to identify concept drifts, and then exploit strategies to handle drifts [37]. Wu et al. [38] proposed a semi-supervised classification algorithm from concept drifting data streams with unlabeled data, named SUN. The SUN constructs an evolved decision tree via a concept drift detection strategy and deviations between historical concept clusters and new clusters generated by the developed k-Modes clustering algorithm. However, SUN cannot directly handle missing features. Theoretically, online learning can naturally deal with concept drifts because of the mechanism, i.e., the model will be updated once an instance with its label is received. You et al. [39] proposed an online Learning model to handle incomplete and imbalanced data streams (OLI^{2}DS). For concept drifts, OLI^{2}DS ensures adaption to unforeseeable changes in the underlying distribution by continuously optimizing the F-measure. However, it mainly focuses on incomplete and imbalanced data and cannot exploit instances with missing labels to update models.

E. Summary

In summary, there are no existing algorithms capable of handling data streams with capricious DIF spaces and random incomplete labels simultaneously. The above approaches cannot effectively handle such a scenario. The details are as follows.

1) For online learning from DIF spaces, a) OLSF [11] and FESL [12] can only process DIF spaces with fixed patterns, i.e., trapezoidal and evolvable feature spaces, and are unsuitable for capricious DIF spaces; b) although OLVF [13] and GLSC [14] can handle capricious DIF spaces, they cannot directly learn from instances with missing labels. Meanwhile, the computational cost is high.
2) For data streams with incomplete labels, a) approaches such as TLP [18] and SPASC [24] cannot learn from incomplete feature spaces; b) approaches such as the model in [25] ignore concept drifts, or identify only recurring drifts, e.g., SPASC [24]; c) approaches such as the model in [25] increase costs because they dynamically update the graphs to arrange similar instance pairs.
3) For SSL, a) approaches such as that in [18], [25], [28], [29] perform poorly on sophisticated data streams because of constraints, e.g., manifold assumption; b) approaches such as those in [30], [31], [32], [33] consume a significant amount of time and space because of the use of ensemble classifiers.
4) To handle concept drifts, a) most approaches such as the model in [36] use detectors to identify concept drifts and design strategies to handle drifts; b) although online learning remains a natural advantage in dealing with concept drifts, existing approaches cannot exploit instances with missing labels to adapt to concept drifts [38], [39].

Based on the above, we propose the OLIFL to efficiently handle data streams with incomplete features and labels, while dynamically adapting to concept drifts.

III. Proposed Algorithm

In this section, a novel classification algorithm for the classification of the **O**nline **L**earning for Data Streams with **I**ncomplete **F**eatures and **L**abels (**OLIFL**) is proposed. OLIFL classifies

data streams with DIF spaces and incomplete labels. The main challenges involve two aspects below.

- Difficulty in learning from capricious DIF spaces. In OLIFL, we introduced the global feature space, which is extended by all observable and disappearing features. Further, the current classifier can learn from global feature space, for which it is weight-updated based on the variation of features.
- Difficulty in updating the classifier using instances without labels. We employ pseudo label [40] and introduce label confidence to exploit the instances without labels. In OLIFL, we estimate label confidence of pseudo label based on the error upper bound, then, instances with low label confidence will be eliminated.

A. Problem Settings

We focus on binary classification tasks because it is easy to convert multi-classification to binary classification sub-tasks using One-vs-Rest [41] or One-vs-One [42] strategies. The adopted symbols and notations are summarized in Table S1 "Symbols and descriptions" in the Supplementary File.

We define $D = \{(x_t, y_t)|t \in \{1, 2, \ldots, T\}\}$ as a sequence of arrival training data, and x_t as an instance arriving at a t-th iteration. Here $x_t \in R^{d_t}$ is a d_t-dimensional feature vector and $y_t \in \{-1, +1\}$ is the associated class label (if available, otherwise, $y_t = \varnothing$) of x_t. At t-th iteration, the model receives x_t and attempts to predict the true label using the function $\hat{y}_t = sign(w_t \cdot x_t)$, where w_t denotes the linear classifier at the t-th iteration. Furthermore, the true label of x_t is revealed and the model suffers an instantaneous loss reflecting the discrepancy between the prediction $\hat{y}_t$ and true label y_t. Additionally, $\mathbb{R}^{g_t} = R^{d_1} \cup R^{d_2} \cup \cdots \cup R^{d_t}$ denotes the global feature space at the t-th iteration that represents the features of $x_1, x_2, \ldots, x_t$ are included, in which $g_t \leq d_1 + d_2 + \cdots + d_t$.

B. Learning From Data Streams With Incomplete Features

For data streams with DIF space, the features of instances are dynamically generated and randomly missing. Due to possible random differences between arriving instances, the feature spaces of any two consecutive instances will result in the degradation of classification performance. Considering the existing approaches, 1) OLSF and FESL can only handle DIF spaces with fixed patterns, i.e., trapezoidal and evolvable feature spaces. 2) Although OLVF and GLSC can handle capricious DIF spaces, the performance of OLVF rapidly degrades as the number of missing features increases because its feature space classifier may dramatically damage the performance of the classifier; GLSC incurs extremely high computational costs due to constructing a generative graphical model and maintaining a latent matrix. Considering the above shortcomings, OLIFL is designed to address DIF space with the following advantages: 1) To reduce the negative impact caused by significant differences between consecutive instances, we construct the global feature space by capturing arriving instances, and obtaining the informativeness of features, thereby a informativeness matrix that reflects the importance of the features; 2) To improve model performance and generalization ability, we partially use unlabeled instances with DIF spaces by restricting the upper bound of errors and the confidence of pseudo labels. Moreover, combined with the proposed informativeness matrix, OLIFL can more effectively cope with concept drift.

The more informativeness an instance has, the better it is for improving the model [43], [44]. If a model is less certain about the prediction on an instance, then the instance is considered to be more informative for improving the model and will be more likely to be selected for label querying. Thus, we propose an adaptive weighing strategy based on feature uncertainty to learn from DIF spaces. At t-th iteration, we set the informativeness matrix as follows:

$$S_t = \begin{bmatrix} e_t^1 & 0 & \cdots & 0 \\ 0 & e_t^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_t^{g_t} \end{bmatrix}_{g_t \times g_t}, \tag{1}$$

where e_t^i is used to measure the informativeness of the feature f_i at t-th iteration. For example, the value of e_1^1 is the informativeness of f_1 at the first iteration. To calculate S_t, the variance vector $v_t = \{a_t^1, a_t^2, \ldots, a_t^{g_t}\}$ is introduced, where a_t^i denotes the variance of f_i, which is obtained iteratively via the variance recursion formula [45]. When a feature appears for the first time, its variances cannot be directly calculated. we set the variance to a positive minimum, e.g., 0.0001, to start learning the feature. As mentioned above, two consecutive instances may have different feature spaces, the feature space can introduce new or old features and some of the existing features may cease to exist. Hence, we define the global feature space as $\mathbb{R}^{g_t} = [f_1, \ldots, f_{d_t}, f^*_{d_{t+1}}, \ldots, f^*_{|g_t|}]$. When an instance is received, the algorithm expands x_t to g_t-dimension by the way that setting all the feature $f_i \in \mathbb{R}^{g_t} \setminus R^{d_t}$ as small scalars (or zeros), which is marked as $\widetilde{x}_t$.

We then calculate the informativeness of each feature in $\mathbb{R}^{g_t}$ using

$$e_t^i = \frac{a_t^i}{\sum_{i=1}^{g_t} a_t^i}, \tag{2}$$

and all e_t^i to form the informativeness matrix S_t.

Furthermore, a feature with more informativeness, i.e., a more significant feature, is updated with a higher weight. When new instances arrive, we first update the $\mathbb{R}^{g_t}$ to find missing features which ensures that missing features are still recorded in $\mathbb{R}^{g_t}$; Then, we use a positive minimum (or zero) to fill the missing feature f_i, which has the following advantages: 1) The vector product of instance x_t and classifier w_t, i.e., prediction result, is constant before and after the fills; 2) A larger variance a_t^i of the feature f_i can bring a higher e_t^i according to (2). Once f_i reappears, it obtains a higher weight, making w_t more favourable to f_i; 3) w_t is updated smoothly because the informativeness of features besides f_i, i.e., $e_t^1 \ldots e_t^{i-1}$ in (4), decreases as the weight of f_i increases.

As instances arrive, w_t is re-weighted based on the informativeness matrix S_t in $\mathbb{R}^{g_t}$ and generates a prediction. We modify

the hinge loss to train a classifier w_t and define it as follows:

$$\ell_t = \max\{0, 1 - y_t \widetilde{w}_t \cdot (\widetilde{x}_t \cdot S_t)^T\}, \tag{3}$$

where $\widetilde{w}_t$ denotes a g_t-dimensional vector, that is expanded from w_t in the same way as x_t to $\widetilde{x}_t$. Furthermore,

$$\widetilde{x}_t \cdot S_t = \begin{bmatrix} f_t^1 \\ f_t^2 \\ \vdots \\ f_t^{g_t} \end{bmatrix}^T \begin{bmatrix} e_t^1 & 0 & \cdots & 0 \\ 0 & e_t^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & e_{g_t}^i \end{bmatrix}. \tag{4}$$

to minimize the cumulative loss in online learning tasks, the weight of an instance is constructed and adapted incrementally after observing each instance, which may be extremely sensitive to noise, causing overfitting and poor generalization. Thus, we use soft-margin with slack variable to design the optimization below.

$$w_{t+1} = \underset{w:\ell(y_t;\hat{y}_t)\leq\xi}{\arg\min} \frac{1}{2}\|w - w_t\|^2 + C\xi, \tag{5}$$

In (5), we assume that an ideal classifier w can classify all instances, then, w_t approximates w through continuous iterations. Equation (5) intends to complete the approximation with minimal changes. To avoid overfitting, ξ is used to constrain the loss in (3) and $C > 0$ to adjust the slackness of the constraint. Thus, with (3) and (5), our learning task can be described as a constrained optimization problem:

$$w_{t+1} = \underset{w:\ell_t\leq\xi,\xi\geq 0}{\arg\min} \frac{1}{2}\|w - \widetilde{w}_t\|^2 + C\xi. \tag{6}$$

To solve (6) we use the Lagrangian function with K.K.T. conditions, and the inequality constraints defined in (6) to obtain the following:

$$L(w, \xi, \tau, \eta) = \frac{1}{2}\|w - \widetilde{w}_t\|^2 + C\xi + \tau(\ell_t - \xi) - \eta\xi, \tag{7}$$

where τ and η denote Lagrange multipliers. To find the solution of $L(w, \xi, \tau, \eta)$, we differentiate $L(w, \xi, \tau, \eta)$ based on $\widetilde{w}_t$ and ξ, to obtain the following conditions:

$$\begin{aligned} w &= \widetilde{w}_t + \tau y_t (\widetilde{x}_t \cdot S_t)^T \\ \eta &= C - \tau. \end{aligned} \tag{8}$$

Substituting (8) into (7), we obtain the following:

$$L(\tau) = \frac{1}{2}\tau^2 \|\widetilde{x}_t \cdot S_t\|^2 + \tau\left(1 - y_t \widetilde{w}_t \cdot (\widetilde{x}_t \cdot S_t)^T\right), \tag{9}$$

$$\tau = \min\left\{C, \frac{\ell_t}{\|\widetilde{x}_t \cdot S_t\|^2}\right\}. \tag{10}$$

The update rule is derived as follows after obtaining the solution to the optimization problem in (6):

$$w_{t+1} = \widetilde{w}_t + \min\left\{C y_t (\widetilde{x}_t \cdot S_t)^T, \frac{\ell_t y_t (\widetilde{x}_t \cdot S_t)^T}{\|\widetilde{x}_t \cdot S_t\|^2}\right\}. \tag{11}$$

C. Learning From Data Streams With Incomplete Features and Labels

Currently, most online learning algorithms assume that labels are available for all received instances. However, missing labels frequently occur in several real-world applications. Furthermore, labeling every instance in data streams is time and resource consuming, making labeling data streams difficult. In reality, only a few instances in data streams have labels, which makes it challenging for existing algorithms to learn from the scenario. After the prediction, the classifier accepts the true label of the instance and then aggressively updates itself based on the loss. However, for an unlabeled instance that does not have a true label as mentioned above, the classifier cannot be updated based on the above strategy.

To overcome the aforementioned drawback, we complete missing labels using pseudo label [40], [46] and introduce the label confidence to filter out instances that negatively affect model updates. The technical novelty of the idea is exhibited as follows: 1) Unlike the models in [40], [46], which are only suitable for offline data, pseudo labels are regarded as the ground truth of unlabeled instances, our method is more suitable for online scenarios by dynamically updating label confidence; 2) We measure the informativeness of unlabeled instances to calculate pseudo labels to extract more effective parts, then, exploit the obtained label confidence to update the classifier to improve model performance; 3) Considering the initial cold start, OLIFL starts learning immediately labeled instances appear. Then, it predicts subsequent unlabeled instances to generate pseudo labels $\overline{y}$ and selects partially unlabeled instances by label confidence to update the classification model. The update mechanisms are as follows.

There is a weight vector w_f for a perceptron that separates the data with a positive margin $\gamma > 0$, such that $y_t(w_f \cdots x_t) > \gamma$ for all $1 \leq t \leq T$ [47]. Then the error upper bound K_M is presented as follows.

$$K_M \leq \frac{R_2 ||w_f||_2}{\gamma}, \tag{12}$$

where $R_2 = MAX_{1\leq t\leq T}\{||x_t||_2\}$ and s represnets the number of the label classes. In particular, we default to M=2, as this is a binary classification problem.

Based on the above theory, we simulate the best linear separator w_f using the current classifier w_t because the w_f cannot be obtained directly by calculations in the streaming data scenario. Then we set $R_2^* = MAX_{1\leq n\leq t}\{||x_n||_2\}$. At t-th iteration that instance missing label, we get the simulated error upper bound k as follows:

$$k \leq \frac{R_2^* ||w_t||_2}{\gamma}. \tag{13}$$

To significantly improve classification performance, we take the maximum value of k in (13) as the theory error of w_t. Furthermore, we denote the number of misclassified instances as m. For w_t, the closeness of the m to the error upper bound reflects how close the current classifier w_t is to w_f, and how close the predicted label is to the true label. Thus, we introduce the label confidence to dynamically measure the credibility of

$\overline{y}$. We define the label confidence as follows:

$$\theta = \frac{m}{k}. \tag{14}$$

Let $\phi(y_t)$ be an activation function defined as follows:

$$\phi\left(y_t\right) = \begin{cases} y_t, & \text{if available} \\ \theta \cdot \overline{y}, & \text{if } y_t = \varnothing. \end{cases} \tag{15}$$

Then, we modify the hinge loss to train a classifier w_t and define it as follows:

$$\ell_t = \max\{0, 1 - \phi\left(y_t\right) \widetilde{w}_t \cdot \left(\widetilde{x}_t \cdot S_t\right)^T\}. \tag{16}$$

The optimization problem is then defined as follows:

$$w_{t+1} = \underset{w:\ell(y_t;\hat{y}_t)\leq\xi}{\arg\min} \frac{1}{2}\|w - w_t\|^2 + C\xi, \tag{17}$$

similar to (6), we solve the optimization problem in (17) and obtain the following:

$$\begin{aligned} w &= \widetilde{w}_t + \tau\phi\left(y_t\right)\left(\widetilde{x}_t \cdot S_t\right)^T, \\ \eta &= C - \tau. \end{aligned} \tag{18}$$

However, we can obtain the following update rules:

$$w_{t+1} = \left[\widetilde{w}_t + \min\left\{C\phi\left(y_t\right)\left(\widetilde{x}_t \cdot S_t\right)^T, \frac{\ell_t\phi\left(y_t\right)\left(\widetilde{x}_t \cdot S_t\right)^T}{\|\widetilde{x}_t \cdot S_t\|^2}\right\}\right]. \tag{19}$$

In summary, we standardize the update rule with (15) for labeled and unlabeled instances. When $\phi(y_t) = y_t$, the data stream degrades to a data stream with incomplete features and updates based on (11) directly, when $\phi(y_t) = \theta \cdot \overline{y}$, we use (19) with θ updates. Furthermore, OLIFL filters partial unlabeled instances that negatively impact the model update by θ based on (13). Specifically, the instance is discarded if the label confidence θ is less than the threshold H, the S_t and w_t are recovered to S_{t-1} and w_{t-1} respectively.

Algorithm 1 depicts the pseudo-code for the OLIFL algorithm.

Algorithm 1: OLIFL Algorithm.

Input:
$C > 0$: Tradeoff parameter of loss
$\gamma > 0$: positive margin
Initialize: $w_1 = (0, 0, \ldots, 0) \in \mathbb{R}^{g_1}$
1: **for** $t = 1, 2, \ldots, T$ **do**
2: Receive instance $x_t \in R^{d_t}$
3: Identify the global feature space $\mathbb{R}^{g_t} = \mathbb{R}^{g_{t-1}} \cup R^{d_t}$
4: Expand w_t, x_t onto the dimension of $\mathbb{R}^{g_t}$: $\widetilde{w}_t, \widetilde{x}_t$
5: Calculate the confidence of features using (2) and composed of S_t
6: Predict the class label $\hat{y}_t = sign(\widetilde{w}_t \cdot (\widetilde{x}_t \cdot S_t))$
7: Receive the true label y_t
8: **if** $y_t = \varnothing$ **then**
9: calculate the maximum k using (13)
10: calculate θ using(14)
11: **if** $\theta < H$ **then**
12: $S_t = S_{t-1}$, $\mathbb{R}^{g_t} = \mathbb{R}^{g_{t-1}}$
13: break
14: **end if**
15: **else if** $y_t \neq \hat{y}$ **then**
16: $m = m + 1$
17: **end if**
18: Suffer loss ℓ_t using (16)
19: Update classifier w_t with $\ell_t, y_t, S_t, \widetilde{x}_t, C$ using (11)
20: **end for**

D. Response to Concept Drifts

OLIFL can adapt to concept drifts in real-time by dynamically updating the classifier w_t as instances arrive. The main mechanisms are as follows.

1) The w_t is updated on line 19 by iteratively executing lines 1–20 in Algorithm 1 and (19) to adapt OLIFL to concept drifts in real-time.

2) The S_t reflects the informativeness of the feature space. Changes of S_t in (1) can render the occurrence of the concept drifts to a certain extent. Therefore, we use S_t as feature weights of (19), which can sensitively respond to drifts.

3) The label confidence θ facilitates the classifier to quickly adapt to drifs. Once drift occurs, the number of misclassifications m increases, which increases θ in (14). From lines 8–16 in Algorithm 1, the increased θ results in more instances with drifts being trained to enable the classifier to adapt more quickly to new distributions.

The above mechanisms make OLIFL more sensitive and precise in adapting to concept drifts. As an extension, when the prediction accuracy falls below a certain threshold, it means that drifts have occurred. OLIFL will be re-trained by receiving new labeled/unlabeled instances on new distributions.

E. Time Complexity

Algorithm 1 shows the pseudo-code of OLIFL. The time complexity for predicting, calculating the informativeness matrix, and suffering losses is $O(|x_t| + |w_t|)$. For a single iteration, the complexity is $O(|x_t| + |w_t|)$, and its runtime is linear. More detailed complexity analysis is provided in Section "S2 COMPLEXITY ANALYSIS" in the Supplementary File.

IV. Theoretical Analysis

In this section, we derive the performance bounds of the OLIFL algorithm. First, we discuss the upper bound of the cumulative hinge loss of OLIFL in a perfect scenario, where a learner can accurately predict each instance, before generalizing and establishing it as linearly inseparable. Finally, we provide OLIFL error rate bounds for each class. Our bounds ensure that our algorithm has a lower cumulative hinge loss than the best-fixed prediction, chosen in hindsight for any sequence of instances. If an instance x_t is falsely predicted, $y_t\widetilde{w}_t \cdot (\widetilde{x}_t \cdot S_t) < 0$, and the loss function $\ell_t > 1$, thus the cumulative squared hinge loss $\sum_t l_t^2$ is an upper bound of the number of false predictions. We denote the loss of the offline predictor at t-th iteration by $\hat{\ell}_t$, and

$\hat{\ell}_t$ is defined as follows:

$$\hat{\ell}_t = \ell\left(\widetilde{\omega}; \left(\widetilde{x}_t, \phi\left(y_t\right) y_t\right)\right) \tag{20}$$

where $\omega \in \mathbb{R}^{g_T}$ represents an arbitrary vector and $\widetilde{x}_t$ represents the expansion of x_t on ω. This notation also applies to $\widetilde{\omega}_t, \widetilde{\omega}_{t+1}$. Thus, we obtain Lemma 1 as follows:

Lemma 1: Let $(x_1, y_1), \ldots, (x_T, y_T)$ be a training data sequence, where $x_t \in R^{d_t}$ and $y_t \in \{+1, -1\}$ for each t. Let the learning rate $\tau = \min\{C, \frac{\ell_t}{\|\widetilde{x}_t \cdot S_t\|^2}\}$, as given in (10). The following bound holds for any $\omega \in \mathbb{R}^{g_T}$

$$\sum_{t=1}^{T} \tau_t (2\ell_t - 2\hat{\ell}_t - \tau_t \|\widetilde{x}_t \cdot S_t\|^2) \leq \|\omega\|^2. \tag{21}$$

The theoretical proof of Lemma 1 is provided in Section "S4 DERIVATIONS AND PROOFS" of the Supplementary File. Lemma 1 proves a loss bound of OLIFL in the perfect case. Specifically, the classifier ω can correctly predict over the sequences, i.e., $y_t(\omega \cdot x_t \cdot S_t) > 0$. By scaling ω, we obtain $y_t(\omega \cdot \widetilde{x}_t \cdot S_t) > 1$, that is, ω can achieve zero hinge loss for all instances over T iterations. Ulteriorly, Theorem 1 is the bound of the cumulative hinge loss of OLIFL.

Theorem 1: Let $(x_1, y_1), \ldots, (x_T, y_T)$ be a sequence of training instances, where $x_t \in R^{d_t}, y_t \in \{+1, -1\}$(if available, otherwise, $y_t = \varnothing$), and $\|x_t\|^2 \leq \mathbb{B}^2$ for all t. Assume that there is a classifier ω such that $\hat{l}_t = 0$ for all t. The cumulative hinge loss of OLIFL over the sequence satisfies

$$\sum_{t=1}^{T} \ell_t \leq \sqrt{\|\omega\|^2 \mathbb{B}^2}. \tag{22}$$

Proof: Because $\hat{l}_t = 0$, based on Lemma 1, we then obtain the following equation:

$$\sum_{t=1}^{T} \tau_t (2\ell_t - \|\widetilde{x}_t \cdot S_t\|^2) \leq \|\omega\|^2. \tag{23}$$

According to the definition $\tau = \frac{\ell_t}{\|\widetilde{x}_t \cdot S_t\|^2}$, we derive the following:

$$\sum_{t=1}^{T} \frac{\ell_t^2}{\|\widetilde{x}_t \cdot S_t\|^2} \leq \|\omega\|^2. \tag{24}$$

Base on the fact that $\|S_t\| = 1$ and $\|x_t\|^2 \leq \mathbb{B}^2$, we have the following:

$$\|\widetilde{x}_t \cdot S_t\|^2 \leq \|x_t\|^2 \tag{25}$$

and

$$\|\widetilde{x}_t \cdot S_t\|^2 \leq \mathbb{B}^2. \tag{26}$$

Substituting (26) into (23), we can obtain

$$\sum_{t=1}^{T} \ell_t \leq \sqrt{\|\omega\|^2 \mathbb{B}^2}. \tag{27}$$

Hence, Theorem 1 is proved. □

Theory 1 demonstrates that 1) an upper bound exists for the cumulative loss in the linearly separable scenario, and 2) the maximum cumulative loss is the right term of the (27), which guarantees that the upper bound is tight.

To obtain the upper bound of the cumulative hinge loss of OLIFL, we generalize the cumulative loss of Theorem 1 to a linearly inseparable scenario in Theorem 2, where the vector $\omega \in \mathbb{R}^{d_T}$ cannot perfectly separate training data.

Theorem 2: Let $(x_1, y_1), \ldots, (x_T, y_T)$ be a sequence of training instances, where $x_t \in R^{d_t}$, $y_t \in \{+1, -1\}$, and $\|x_t\|^2 = 1$ for all t. The following bound holds for any vector $\omega \in \mathbb{R}^{d_T}$, which is the cumulative hinge loss of OLIFL over the sequence:

$$\sum_{t=1}^{T} \ell_t \leq \sqrt{\|\omega\|^2 + 2\sum_{t=1}^{T} \hat{l}_t \ell_t}. \tag{28}$$

Proof: Since $\|x_t\|^2 = 1$, and $\tau = \frac{\ell_t}{\|\widetilde{x}_t \cdot S_t\|^2}$, based on Lemma 1, we then deduce:

$$\sum_{t=1}^{T} \frac{2\ell_t^2 - 2\hat{l}_t \ell_t - \ell_t^2}{\|\widetilde{x}_t \cdot S_t\|^2} \leq \|\omega\|^2. \tag{29}$$

From (25) we have

$$\|\widetilde{x}_t \cdot S_t\|^2 \leq 1. \tag{30}$$

According to (29) and (30), we have

$$\sum_{t=1}^{T} \ell_t^2 - \sum_{t=1}^{T} 2\hat{l}_t \ell_t \leq \sum_{t=1}^{T} \frac{\ell_t^2 - 2\hat{l}_t \ell_t}{\|\widetilde{x}_t \cdot S_t\|^2} \leq \|\omega\|^2. \tag{31}$$

After rearranging (31), we can obtain

$$\sum_{t=1}^{T} \ell_t^2 \leq \|\omega\|^2 + \sum_{t=1}^{T} 2\hat{l}_t \ell_t, \tag{32}$$

then, the cumulative first-order hinge losses are obtained as follows,

$$\sum_{t=1}^{T} \ell_t \leq \sqrt{\|\omega\|^2 + 2\sum_{t=1}^{T} \hat{l}_t \ell_t}. \tag{33}$$

Hence, Theorem 2 is proved. □

Theorem 2 demonstrates that: 1) the cumulative hinge loss of (16) is not infinite with the first-order bound, which guarantees the theoretical upper bound under linear and inseparable conditions; 2) The maximum of first-order cumulative loss is the right term of (33), remaining a tight upper bound.

Considering the upper bound of cumulative loss in Theorem 2, we construct Theorem 3 to analyze an error upper bound K_M.

Theorem 3: Let $(x_1, y_1), \ldots, (x_T, y_T)$ be a sequence of training instances, where $x_t \in R^{d_t}$, $y_t \in \{+1, -1\}$, and $\|x_t\| \leq \mathbb{B}$ for all t. The following bound holds for any vector $\omega \in \mathbb{R}^{d_T}$, which is the number of misclassifications of OLIFL on any class over the sequence:

$$K_M \leq \max\left\{\frac{1}{C}, \mathbb{B}^2\right\}\left(\|\omega\|^2 + 2C\sum_{t=1}^{T} \hat{l}_t\right). \tag{34}$$

Proof: As previously mentioned, the cumulative hinge loss is an upper bound of the number of misclassified instances. We

thus have

$$\sum_{s=1}^{S} K_M \leq \sum_{t=1}^{T} \ell_t, \tag{35}$$

K_M is the number of misclassified instances for any class, such as K_1 and K_2, representing the number of misclassified instances for $y_t = 1$ and $y_t = 2$, respectively, where M is the number of the labels' classes. Combining the definition $\tau = \frac{\ell_t}{\|\widetilde{x}_t \cdot S_t\|^2}$ and (35), we obtain the following:

$$\min\left\{C, \frac{1}{\|\widetilde{x}_t \cdot S_t\|^2}\right\} K_M \leq \sum_{t=1}^{T} \tau_t \ell_t. \tag{36}$$

From the definition of τ_t, we have obtained $\tau_t \hat{l}_t \leq C\hat{l}_t$ and $\tau_t \|\widetilde{x}_t \cdot S_t\|^2 \leq \ell_t$. Using these two inequalities and Lemma 1, we can obtain the following:

$$\sum_{t=1}^{T} \tau_t \ell_t \leq \|\omega\|^2 + 2C \sum_{t=1}^{T} \hat{l}_t. \tag{37}$$

Combining (36) and (37), we obtain the following equation:

$$\min\left\{C, \frac{1}{\|\widetilde{x}_t \cdot S_t\|^2}\right\} K_M \leq \|\omega\|^2 + 2C \sum_{t=1}^{T} \hat{l}_t. \tag{38}$$

After rearranging (38), the follow can be obtained:

$$K_M \leq \max\left\{\frac{1}{C}, \|\widetilde{x}_t \cdot S_t\|^2\right\} \left(\|\omega\|^2 + 2C \sum_{t=1}^{T} \hat{l}_t\right). \tag{39}$$

Based on $\|x_t\| \leq \mathbb{B}$, (26) and (39), we can obtain the number of misclassifications of OLIFL on any class as follow:

$$K_M \leq \max\left\{\frac{1}{C}, \mathbb{B}^2\right\} \left(\|\omega\|^2 + 2C \sum_{t=1}^{T} \hat{l}_t\right). \tag{40}$$

Hence, Theorem 3 is proved. □

Theorem 3 presents misclassified instances of classifier ω at t-th iteration is finite and exists an error upper bound. For (40), 1) due to the maximum of K_M being its right item, no upper bound is better than the right term. Therefore, the upper bound of K_M in (40) is tight; 2) The tight bound ensures that K_M can be maximized, then the number of misclassifications stops rising, and ω achieves ideal performance.

V. Experiments

To cover more general and challenging cases of online learning from DIF spaces with missing labels, we adopt the following experiments: 1) analyzing performance on DIF spaces with complete/incomplete labels; 2) evaluating the adaptability to concept drifts, and 3) verifying the usability and response efficiency of OLIFL.

A. General Settings

Datasets and Parameters: To evaluate performance, we perform experiments on 9 real datasets, a real-world large-scale dataset, and 9 large-scale synthetic datasets generated by MOA [48] generator. Table II summarizes the statistics of the datasets. Specifically, 1) To analyze DIF with complete labels, we remove partial features from 12 datasets to construct trapezoidal, evolvable, and capricious streams, respectively; 2) To evaluate DIF with incomplete labels, we randomly remove some features and labels with different FR and LR; 3) To evaluate the adaptability of OLIFL to concept drifts, we generate 2 datasets by MOA with sudden and gradual drifts, respectively; 4) To analyze the usability of OLIFL, we simulate 4 datasets with different FR, LR, and drifts using MOA. The details of the settings of datasets and parameters are shown in Section "S4.1 Implementation Details" of the Supplementary File.

TABLE I
Summary of Compared Algorithms

Algorithms	Patterns of Feature Space				Label Attribute	
	IFs			CFs	CL	IL
	TDR	FES	CDS			
OLSF [11]	✓			✓	✓	
FESL [12]		✓		✓	✓	
OLVF [13]	✓	✓	✓	✓	✓	
GLSC [14]	✓	✓	✓	✓	✓	
LNP [34]					✓	✓
REAL [19]					✓	✓
OLIFL	✓	✓	✓	✓	✓	✓

*The CFs/IFs, CL/IL denote the abbreviations of Complete/Incomplete Features and Complete/Incomplete Labels, respectively. The TDR, FES and CDS denote Trapezoidal Data Stream, Feature Evolvable Stream, and Capricious Data Stream, respectively.

Evaluation Metrics: We use accuracy and runtime as evaluation indicators. Then, we analyze the average classification accuracy of OLIFL with different FR, LR, and concept drifts. In detail, we use the test-then-train [49] pattern on datasets with random instance sequences repeated 10 times. To observe the trend of performance changes on datasets with concept drifts, we only run the datasets with original instance sequences by repeating them 10 times.

Baselines: As shown in Table I, we compare OLIFL with OLSF, FESL, OLVF, and GLSC in trapezoidal, feature evolvable, and capricious data streams, respectively. To better evaluate the performance of OLIFL on handling data streams with incomplete features and labels, we select two semi-supervised algorithms for data streams, i.e., LNP and REAL, as baselines.

B. Experiments on Incomplete Features and Complete Labels

In this section, we compare our OLIFL with four algorithms of handling DIF with complete labels on trapezoidal, feature evolvable, and capricious data streams, respectively.

1) Experiments on Trapezoidal Data Streams: In this section, we compare OLIFL with OLSF [11] for handling trapezoidal data streams. Table III shows that OLIFL achieves an average improvement of 0.1 over OLSF on all datasets and outperforms OLSF in terms of runtime under the same settings. This is because OLIFL considers the variation in individual features when determining feature confidence, whereas OLSF does not; thus, OLIFL is superior. Although OLIFL considers more features, it does not require feature sparse and has lower

TABLE II
STATISTICS OF THE DATASETS IN OUR EXPERIMENTS

Source	Datasets	Instance	Dims	Datasets	Instance	Dims
real world	australian	690	14	splice	3190	60
	credit-a	690	15	spambase	4601	57
	dna	949	180	magic	19020	10
	credit-g	1000	20	hyper_f	1000000	10
	german	1000	24	IMDB	25000	7500

Source	Datasets	Instance	Dims	Drift Range	Datasets	Instance	Dims
generated by MOA	sudden	200000	1000	[40000,40001]	moa1	10000	1000
	gradual	200000	1000	[50000,70000]			
	S1	10000	100	[5000,5001]	moa2	100000	100
	G1	10000	100	[5000,6000]			
	S1G1	10000	100	[2000,2001],[4000,5000]	moa3	100000	1000
	G1S1	10000	100	[2000,3000],[4000,4001]			

TABLE III
EXPERIMENTAL RESULTS (MEAN ACCURACY ± STANDARD DEVIATION, MEAN RUNTIME ± STANDARD DEVIATION) ON 12 DATA SETS IN SIMULATED TRAPEZOIDAL DATA STREAMS

	Accuracy		Runtime(s)	
Dataset	OLSF	OLIFL	OLSF	OLIFL
australian	.598±.033	**.670±.015**	.813±.007	**.044±.001**
credit-a	.572±.028	**.620±.012**	.774±.003	**.040±.001**
dna	.686±.016	**.771±.008**	1.362±.006	**.465±.007**
credit-g	.532±.016	**.670±.010**	1.374±.009	**.087±.001**
german	.553±.023	**.676±.011**	1.478±.007	**.085±.001**
splice	.622±.006	**.647±.005**	8.311±0.051	**.597±.004**
spambase	.747±.019	**.793±.011**	16.256±.068	**.767±.018**
magic	.645±.005	**.694±.003**	234.624±5.978	**1.163±.006**
moa1	.769±.001	**.884±.001**	54.914±0.3418	**3.069±.080**
moa2	.769±.001	**.877±.000**	56.804±1.230	**2.803±.012**
moa3	.770±.001	**.903±.000**	6515.225±206.866	**28.525±.223**
hyper_f	.645±.005	**.775±.036**	9515.325±115.978	**57.901±.153**

The bold values indicate that the corresponding algorithm outperforms its rivals.

TABLE IV
EXPERIMENTAL RESULTS (ACCURACY ± STANDARD DEVIATION, RUNTIME ± STANDARD DEVIATION) ON 12 DATA SETS IN SIMULATED FEATURE EVOLVABLE STREAMS

	Accuracy		Runtime(s)	
Dataset	FESL	OLIFL	FESL	OLIFL
australian	**.830±.023**	.822±.024	**.214±.001**	.330±.002
credit-a	.767±.227	**.818±.022**	**.203±.001**	.445±.002
dna	.795±.047	**.933±.007**	**.321±.004**	6.127±.018
credit-g	.641±.020	**.652±.011**	**.312±.002**	.863±.005
german	.640±.017	**.655±.011**	**.302±.004**	.737±.006
splice	.693±.023	**.778±.011**	**1.041±.012**	5.585±0.135
spambase	.693±.023	**.778±.011**	**1.025±.006**	5.734±.030
magic	.799±.037	**.880±.007**	**1.428±.017**	10.098±.040
moa1	.843±.013	**.894±.001**	**28.993±0.277**	35.694±1.204
moa2	.846±.027	**.898±.012**	**36.579±.041**	28.993±.277
moa3	.893±.023	**.913±.037**	**33.420±0.938**	45.694±1.204
hyper_f	.767±.067	**.857±.034**	**117.110±11.354**	139.470±6.003

The bold values indicate that the corresponding algorithm outperforms its rivals.

time complexity, which explains its superiority over OLSF in terms of runtime.

2) Experiments on Feature Evolvable Streams: In this section, we compare OLIFL with FESL [12] for handling feature evolvable streams. As shown in Table IV, OLIFL achieves higher accuracy than FESL on 11 datasets excluding *austrilian*. This is because FESL learns a fixed mapping matrix during the period when old and new features coexist. On the other hand, OLIFL is trained under the global feature space, which brings more information. OLIFL sacrifices runtime for better classification accuracy by maintaining the global feature space to calculate feature weights, thus FESL performs slightly better in terms of runtime.

3) Experiments on Capricious Data Streams: In this section, we compare OLIFL with OLVF [13] and GLSC [14] for handling capricious data streams. Table V shows that OLIFL outperforms OLVF and GLSC in terms of runtime on all 12 datasets. Additionally, OLIFL significantly outperforms OLVF and GLSC in accuracy on almost all datasets. The results indicate that OLIFL can efficiently handle capricious feature spaces. This is because it considers the variation of individual features, which simplifies the computation. However, OLVF only performs a coarse division of the feature space; and GLSC requires more features to classify based on the graphical model for partitioning, which explains its far inferior performance to ours in runtime.

C. Experiments on Complete Features and Incomplete Labels

In this section, we compare OLIFL with two semi-supervised algorithms, i.e., online REAL [23] and offline LNP [34]. With the results in Fig. 1, we observe that OLIFL achieves different accuracy improvements compared with REAL and LNP on almost all datasets at different LR. This is because OLIFL dynamically computes the label confidence, and eliminates instances that negatively affect updating the model. The results indicate that OLIFL can efficiently handle data streams with incomplete labels. Additionally, Fig. 2 shows OLIFL maintains a stable performance on multiple datasets with different γ ($10^{-3} \sim 1$) and LR, which proves the robustness of OLIFL.

D. Experiments on Incomplete Features and Labels

In this section, we evaluate OLIFL and its variants on data streams with incomplete features and labels. We set the default FR$=$ 0.5 and LR $=$ 0.99, 0.95, 0.9, 0.8, and 0.7. Fig. 3 shows the trends of accuracy with iterations on the 12 datasets. We can observe that the accuracy improves rapidly and gradually maintains a more consistent value even among different LR. Remarkably, the classification accuracy shows a wider range of improvement at a high LR; and at an extremely low LR, OLIFL struggles to learn from the only available labeled data and shows effectiveness. Ulteriorly, we compare OLIFL with its variant OLIFL-u, which ignores the missing label instances,

TABLE V
EXPERIMENTAL RESULTS (MEAN ACCURACY ± STANDARD DEVIATION, MEAN RUNTIME ± STANDARD DEVIATION) ON 12 DATA SETS IN SIMULATED DATA STREAMS WITH INCOMPLETE FEATURES

	Accuracy			Runtime(s)		
Dataset	GLSC	OLVF	OLIFL	GLSC	OLVF	OLIFL
australian	.648±.085	.676±.029	**.786±.019**	2.279±1.321	.836±.013	**.097±.001**
credit-a	.667±.134	.668±.018	**.782±.020**	2.236±1.294	.870±.011	**.099±.001**
dna	.529±.056	.772±.019	**.883±.017**	16.705±9.395	2.126±.010	**1.329±.012**
credit-g	.617±.033	.562±.011	**.622±.009**	3.738±2.170	1.554±.010	**.187±.002**
german	**.630±.038**	.566±.021	.616±.016	4.177±2.433	1.742±.006	**.214±.005**
splice	.509±.020	.634±.006	**.728±.008**	21.449±12.545	9.349±0.056	**1.582±.006**
spambase	.670±.043	.760±.007	**.807±.018**	31.970±19.221	18.692±.113	**2.148±.007**
magic	.687±.018	.587±.006	**.692±.003**	123.939±92.971	216.561±3.353	**2.091±.011**
moa1	.651±0.052	.804±0.008	**.917±.004**	38.692±24.151	33.854±0.066	**3.708±0.030**
moa2	.732±0.045	.804±0.004	**.934±.001**	1476.172±1262.639	2942.228±27.959	**37.915±0.434**
moa3	.762±0.005	.802±0.001	**.984±.001**	1432.806±1224.519	2963.417±1.896	**40.031±.225**
hyper_f	.697±.013	.698±.007	**.722±.004**	3324.346±1132.001	216.561±3.353	**112.500±3.611**

The bold values indicate that the corresponding algorithm outperforms its rivals.

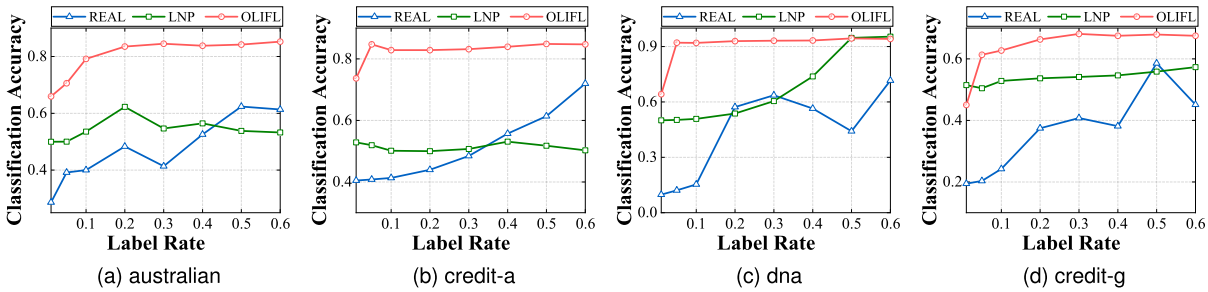

Fig. 1. Trends in the average classification accuracy of REAL, LNP, and OLIFL on different Label Rate, wherein Label Rate=1-LR. Due to the page limits, all 12 datasets are shown in Fig. S1 of the Supplementary File.

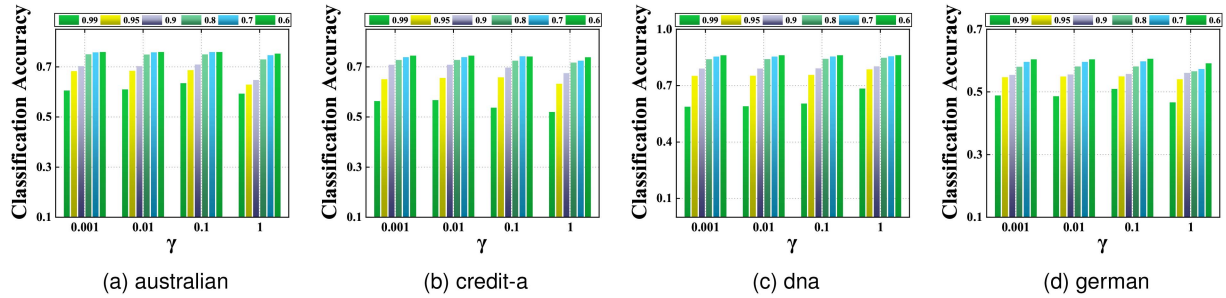

Fig. 2. Impacts of positive margin γ on the accuracy of OLIFL on different LR, wherein LR=0.99, 0.95, 0.9, 0.8, 0.7, 0.6. Due to the page limits, all 12 datasets are shown in Fig. S2 of the Supplementary File.

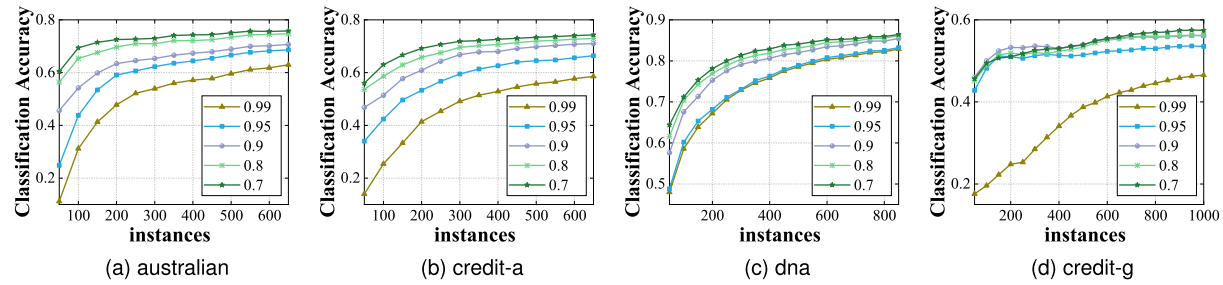

Fig. 3. Trends of average classification accuracy of OLIFL with different LR when FR=0.5, wherein LR=0.99, 0.95, 0.9, 0.8, 0.7. Due to the page limits, all 12 datasets are shown in Fig. S3 of the Supplementary File.

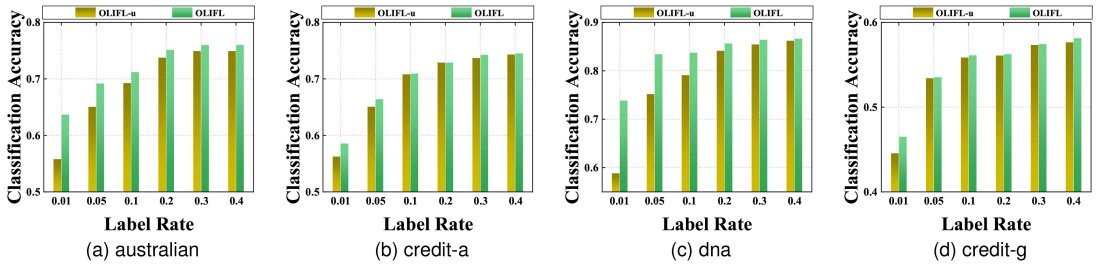

Fig. 4. Comparison of OLIFL and its variant OLIFL-u on 12 datasets with different Label Rate, wherein Label Rate=1-LR. Due to the page limits, all 12 datasets are shown in Fig. S4 of the Supplementary File.

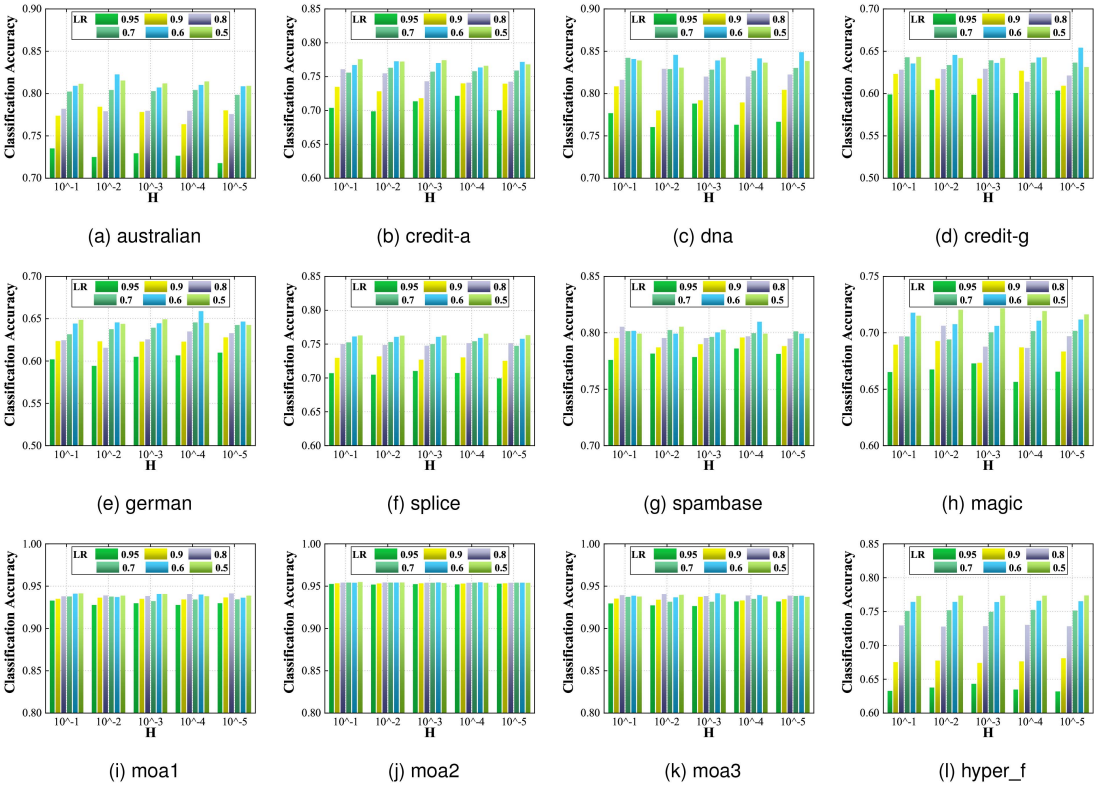

Fig. 5. Trends of average classification accuracy of OLIFL with different H and LR under FR=0.5.

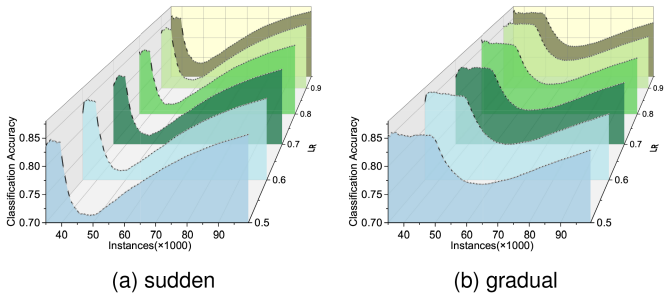

Fig. 6. Trends of average classification accuracy of OLIFL with different LR and sudden/gradual concept drift when FR=0.5.

under different values of LR, as shown in Fig. 4. We can see that the label weighting and the elimination strategies effectively improve the classification accuracy, particularly when the data stream has a lower value of LR. To measure the impact of parameter H on OLIFL, we search for different values of H with LR = 0.95, 0.9, 0.8, 0.7, 0.6, or 0.5, FR= 0.5. As shown in Fig. 5, the fluctuation is small at $H = 0.0001$ with LR ranging from 0.95 to 0.5. The ideal range is approximately 0.0001. This is because the value of H depends on the training data; a lower H can lead to reduced utilization of unlabeled instances, resulting in information waste. Conversely, a higher H can trigger the incorporation of redundant unlabeled instances, consequently contributing to performance degradation.

E. Experiments on Incomplete Features and Labels With Concept Drift

In this section, we evaluate the adaptability of OLIFL to concept drifts. We generate datasets with sudden and gradual concept drifts by MOA, respectively. For sudden drift, we set

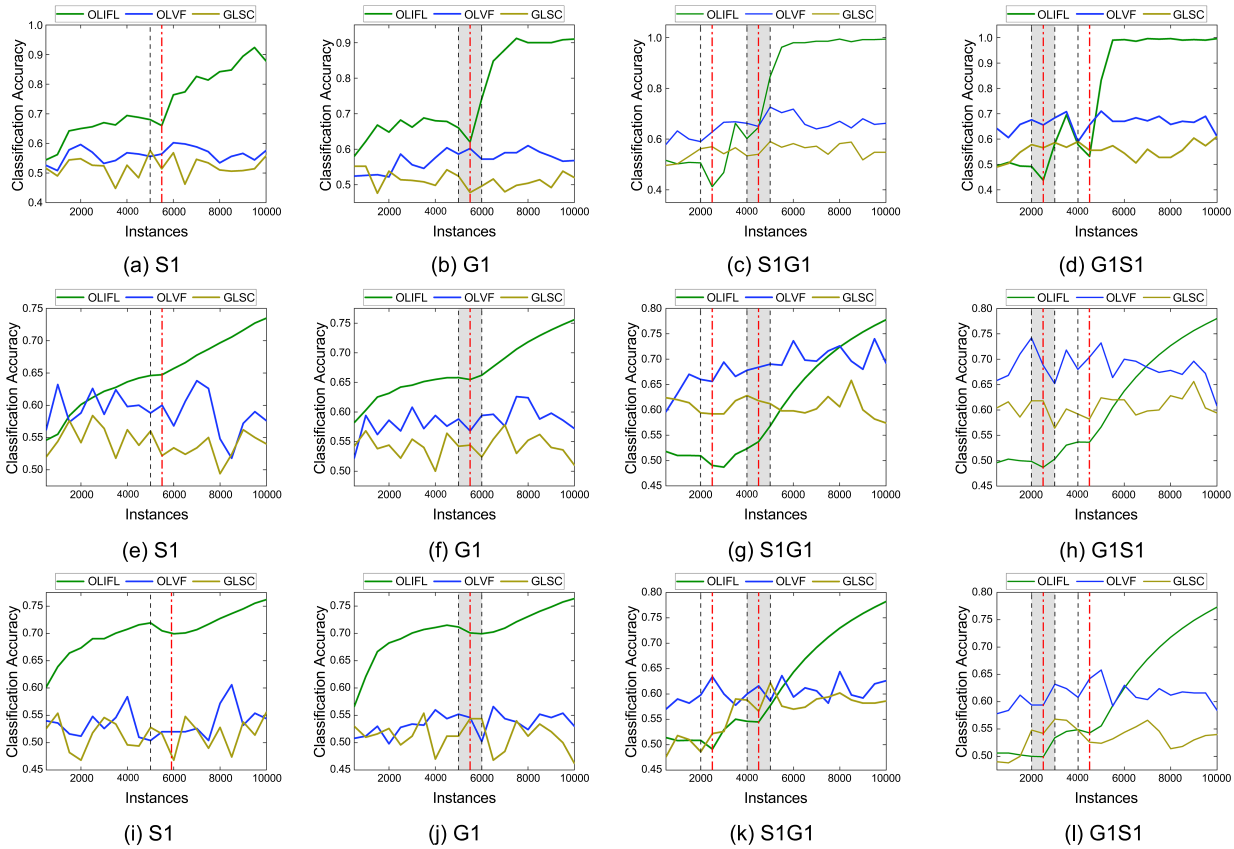

Fig. 7. Usability and response efficiency of OLIFL and its rivals with the arrival of instances under different concept drifts, FR, and LR. Each instance iteration in the x-axis represents an increase in unit time. The symbols S1, G1, S1G1, and G1S1 denote the different drifts: sudden, gradual, sudden → gradual, and gradual→ sudden,respectively; (a)–(d): FR=0.5, LR = 0.5; (e)–(h): FR=0.7, LR = 0.3; (i)–(l): FR=0.3, LR = 0.7.

the drift to occur at i = 40000. For the gradual drift, we set the drift to occur at i = 50000 with a window size of 20000. Furthermore, we set FR = 0.5 and LR = 0.95, 0.9, 0.8, 0.7, and 0.6. As shown in Fig. 6(a) and (b), 1) when confronted with sudden concept drift, the average accuracy has more significant fluctuations, because the data distribution changes abruptly, while in gradual concept drift the data distribution changes gradually; 2) When the concept drift does not occur, the average accuracy gradually increases. When the concept drift occurs (i=40,000 sudden and i=50,000 in gradual respectively), the average accuracy decreases gradually, then rebounds rapidly after reaching the lowest point, finally achieving a stable and excellent performance. This is because OLIFL can sensitively calculate the variance changes of features based on the global feature space to update the feature weights, which change significantly when concept drift occurs and thus better address drifts. The above results demonstrate that OLIFL can handle concept drift as mentioned in Section III-D.

F. Usability and Response Efficiency

In this section, we analyze the usability by performing data streams with different FR, LR, and concept drifts. Meanwhile, we observe the response efficiency in handling concept drifts.

Usability: From Fig. 7, we can note that 1) under different FR and LR, OLIFL significantly outperforms its rivals when the type of drifts is identical, e.g., (a),(e), and (i) belong to S1; 2) Under the same LR and FR, e.g., (a)–(d), OLIFL eventually outperforms its rivals and maintains fewer fluctuations on different drifts, especially when handling single-type drift, e.g., (a) and (b); 3) For complex concept drifts, e.g., S1G1 and G1S1, OLIFL exhibits positive usability with stable performance improvements, as shown in Fig. 7(c)–(d), (g)–(h), and (k)–(i). Overall, for any FR, LR, and concept drift in Fig. 7(a)–(l), OLIFL maintains a continuous rise in performance and tends to stabilize as instances continue to arrive. Therefore, our OLIFL is suitable for data streams with different FR, LR, and concept drifts because it updates the classifier in real-time, and employs the strategies of informativeness matrix (1) and label confidence (14) which ensure that additional information is extracted from data streams.

Response efficiency to concept drifts: To observe the response efficiency of OLIFL when concept drifts occur, as shown in Fig. 7, we set each iteration of instances on the x-axis to represent an increase in unit time. The black short dash indicates the occurrence/end of drifts, the shadow indicates the range of gradual drift, and the red dot dash indicates that OLIFL has adapted to the drifts. As shown in Fig. 7, 1) under the single type drifts, i.e., S1 and G1, the accuracy of OLIFL can rapidly

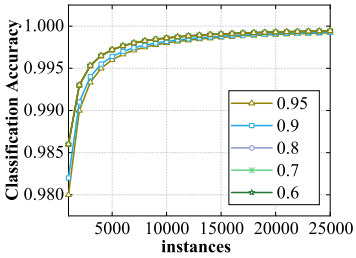

Fig. 8. Trends of average classification accuracy of OLIFL in a real-world dataset with respect to different values of LR when FR=0.5.

recover at approximately i = 5500 (red dot-dash) after drops due to the drifts at i = 5000 (black dash), regardless of the values of FR and LR. On this issue, OLIFL outperforms its rivals; 2) under the complex drifts, i,e., S1G1, and G1S1, OLIFL does not exhibit excellent performance (sudden drift when i=2000, black dash) when i<2500 (red dot-dash) because evaluating a few unlabeled instances with drifts can result in errors. However, as shown in the curves in (c)-(d), (g)-(h), and (k)-(i), once the number of instances i> 4000, even if there are different types of drift (i = 4000, gradual in (c), (g) and (k), sudden in (d), (h) and (i)), OLIFL starts to recover at i=4500 (red dot-dash) and quickly responds to improve classification accuracy. Overall, OLIFL eventually outperforms its rivals in response efficiency to concept drifts because of the mechanisms in subsection III-D.

VI. A Case Study on Movie Review

In this section, we apply OLIFL to a real-world scenario to verify its effectiveness. We use the large-scale movie review dataset IMDB[1] as a benchmark [50]. IMDB contains 50,000 movie reviews, where the ratings are given as star values ($\in \{1, 2, 3, \ldots, 10\}$). In the classification task, movie reviews are generated continuously in the form of streams. Each movie review consists of different words, and each word can be seen as a feature. We treat half of the star values as positive and the other as negative. We set FR=0.5 and LR=0.95, 0.9,..., 0.5.

Fig. 8 shows that OLIFL can achieve robust and stable classification performance in real-world applications. This is because OLIFL dynamically updates the individual feature weights based on the global feature space, which effectively exploits the information of the surviving features only; the label confidence based on error upper bounds exploits information about instances with missing labels, and the elimination strategy avoids the negative impact of overlearning unlabeled instances.

VII. Conclusion

In this study, we proposed the OLIFL to handle DIF spaces with incomplete labels. We exploited specific changes of a single feature to obtain its informativeness by maintaining the global feature space including old and new features. Furthermore, we updated the classifier by limiting the error upper bound to assign a confidence level to the predicted labels. Extensive experiments demonstrated that the OLIFL algorithm outperforms its rivals on most datasets. In addition, OLIFL is outstanding under different patterns of streaming data, which implies superior generalization ability and robustness.

In the future, we will study how to online learning from DIF space with missing and constantly emerging new labels and constantly emerging new class labels [51], and design a more generalized classifier with adaptive parameter updates.

[1] https://www.imdb.com

References

[1] H. Zhao, H. Wang, Y. Fu, F. Wu, and X. Li, "Memory efficient class-incremental learning for image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 10, pp. 5966–5977, Oct. 2022.
[2] K. Yu, L. Liu, J. Li, W. Ding, and T. D. Le, "Multi-source causal feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 9, pp. 2240–2256, Sep. 2020.
[3] N. Suchetha, A. Nikhil, and P. Hrudya, "Comparing the wrapper feature selection evaluators on twitter sentiment classification," in *Proc. Int. Conf. Comput. Intell. Data Sci.*, 2019, pp. 1–6.
[4] K. Yu, L. Liu, and J. Li, "A unified view of causal and non-causal feature selection," *ACM Trans. Knowl. Discov. Data*, vol. 15, no. 4, pp. 1–46, 2021.
[5] Y. Zhou, G. Cheng, S. Jiang, and M. Dai, "Building an efficient intrusion detection system based on feature selection and ensemble classifier," *Comput. Netw.*, vol. 174, 2020, Art. no. 107247.
[6] V. Verma and R. K. Aggarwal, "A comparative analysis of similarity measures akin to the jaccard index in collaborative recommendations: Empirical and theoretical perspective," *Social Netw. Anal. Mining*, vol. 10, no. 1, pp. 1–16, 2020.
[7] D. Wu, Y. He, X. Luo, and M. Zhou, "A latent factor analysis-based approach to online sparse streaming feature selection," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 11, pp. 6744–6758, Nov. 2022.
[8] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with Big Data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 97–107, Jan. 2014.
[9] Y. Wang et al., "Novelty detection and online learning for chunk data streams," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2400–2412, Jul. 2021.
[10] Y. Song, J. Lu, H. Lu, and G. Zhang, "Learning data streams with changing distributions and temporal dependency," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 8, pp. 3952–3965, Aug. 2023.
[11] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang, and X. Wu, "Online learning from trapezoidal data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2709–2723, Oct. 2016.
[12] B.-J. Hou, L. Zhang, and Z.-H. Zhou, "Learning with feature evolvable streams," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 6, pp. 2602–2615, Jun. 2021.
[13] E. Beyazit, J. Alagurajah, and X. Wu, "Online learning from data streams with varying feature spaces," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3232–3239.
[14] Y. He, B. Wu, D. Wu, E. Beyazit, S. Chen, and X. Wu, "Toward mining capricious data streams: A generative approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1228–1240, Mar. 2021.
[15] M. M. Masud et al., "Facing the reality of data stream classification: Coping with scarcity of labeled data," *Knowl. Inf. Syst.*, vol. 33, no. 1, pp. 213–244, 2012.
[16] J. Shao, C. Huang, Q. Yang, and G. Luo, "Reliable semi-supervised learning," in *Proc. IEEE 16th Int. Conf. Data Mining*, 2016, pp. 1197–1202.
[17] R. S. Ferreira, G. Zimbrão, and L. G. Alvim, "Amanda: Semi-supervised density-based adaptive model for non-stationary data with extreme verification latency," *Inf. Sci.*, vol. 488, pp. 219–237, 2019.
[18] T. Wagner, S. Guha, S. Kasiviswanathan, and N. Mishra, "Semi-supervised learning on data streams via temporal label propagation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5095–5104.
[19] Y. He, X. Yuan, S. Chen, and X. Wu, "Online learning in variable feature spaces under incomplete supervision," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 4106–4114.
[20] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Comput. Surv.*, vol. 46, no. 4, pp. 1–37, 2014.

[21] Z. Li, W. Huang, Y. Xiong, S. Ren, and T. Zhu, "Incremental learning imbalanced data streams with concept drift: The dynamic updated ensemble algorithm," *Knowl.-Based Syst.*, vol. 195, 2020, Art. no. 105694.
[22] J. L. Lobo, J. Del Ser, A. Bifet, and N. Kasabov, "Spiking neural networks and online learning: An overview and perspectives," *Neural Netw.*, vol. 121, pp. 88–100, 2020.
[23] S. U. Din, J. Shao, J. Kumar, W. Ali, J. Liu, and Y. Ye, "Online reliable semi-supervised learning on evolving data streams," *Inf. Sci.*, vol. 525, pp. 153–171, 2020.
[24] M. J. Hosseini, A. Gholipour, and H. Beigy, "An ensemble of cluster-based classifiers for semi-supervised classification of non-stationary data streams," *Knowl. Inf. Syst.*, vol. 46, no. 3, pp. 567–597, 2016.
[25] L. Zhu, S. Pang, A. Sarrafzadeh, T. Ban, and D. Inoue, "Incremental and decremental max-flow for online semi-supervised learning," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2115–2127, Aug. 2016.
[26] W. Shao, Z. Ge, and Z. Song, "Semisupervised bayesian gaussian mixture models for non-gaussian soft sensor," *IEEE Trans. Cybern.*, vol. 51, no. 7, pp. 3455–3468, Jul. 2021.
[27] G. Ditzler and R. Polikar, "Semi-supervised learning in nonstationary environments," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 2741–2748.
[28] Y. Chong, Y. Ding, Q. Yan, and S. Pan, "Graph-based semi-supervised learning: A review," *Neurocomputing*, vol. 408, pp. 216–230, 2020.
[29] Z. Liu, Z. Lai, W. Ou, K. Zhang, and R. Zheng, "Structured optimal graph based sparse feature extraction for semi-supervised learning," *Signal Process.*, vol. 170, 2020, Art. no. 107456.
[30] Z. Ahmadi and H. Beigy, "Semi-supervised ensemble learning of data streams in the presence of concept drift," in *Proc. Int. Conf. Hybrid Artif. Intell. Syst.*, 2012, pp. 526–537.
[31] J. Peng, G. Estrada, M. Pedersoli, and C. Desrosiers, "Deep co-training for semi-supervised image segmentation," *Pattern Recognit.*, vol. 107, 2020, Art. no. 107269.
[32] P. Wang, J. Peng, M. Pedersoli, Y. Zhou, C. Zhang, and C. Desrosiers, "Self-paced and self-consistent co-training for semi-supervised image segmentation," *Med. Image Anal.*, vol. 73, 2021, Art. no. 102146.
[33] Y. Wang and T. Li, "Improving semi-supervised co-forest algorithm in evolving data streams," *Appl. Intell.*, vol. 48, no. 10, pp. 3248–3262, 2018.
[34] F. Wang and C. Zhang, "Label propagation through linear neighborhoods," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 1, pp. 55–67, Jan. 2008.
[35] H. Mehmood, P. Kostakos, M. Cortes, T. Anagnostopoulos, S. Pirttikangas, and E. Gilman, "Concept drift adaptation techniques in distributed environment for real-world data streams," *Smart Cities*, vol. 4, no. 1, pp. 349–371, 2021.
[36] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 12, pp. 2346–2363, Dec. 2019.
[37] D. Brzezinski and J. Stefanowski, "Reacting to different types of concept drift: The accuracy updated ensemble algorithm," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 81–94, Jan. 2014.
[38] X. Wu, P. Li, and X. Hu, "Learning from concept drifting data streams with unlabeled data," *Neurocomputing*, vol. 92, pp. 145–155, 2012.
[39] D. You et al., "Online learning from incomplete and imbalanced data streams," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 10, pp. 10650–10665, Oct. 2023.
[40] Z. Zheng and Y. Yang, "Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 1106–1120, 2021.
[41] J. Xu, "An extended one-versus-rest support vector machine for multi-label classification," *Neurocomputing*, vol. 74, no. 17, pp. 3114–3124, 2011.
[42] J. A. Sáez, M. Galar, J. Luengo, and F. Herrera, "Analyzing the presence of noise in multi-class problems: Alleviating its influence with the one-vs-one decomposition," *Knowl. Inf. Syst.*, vol. 38, no. 1, pp. 179–206, 2014.
[43] S.-J. Huang, M. Xu, M.-K. Xie, M. Sugiyama, G. Niu, and S. Chen, "Active feature acquisition with supervised matrix completion," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 1571–1579.
[44] S.-J. Huang, R. Jin, and Z.-H. Zhou, "Active learning by querying informative and representative examples," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 10, pp. 1936–1949, Oct. 2014.
[45] H. Patterson, "Sampling on successive occasions with partial replacement of units," *J. Roy. Stat. Society. Ser. B Methodological*, vol. 12, no. 2, pp. 241–255, 1950.
[46] X. Gu, J. Sun, and Z. Xu, "Spherical space domain adaptation with robust pseudo-label loss," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9101–9110.
[47] F. Martin, B. Stamper, and C. Flowers, "Examining student perception of readiness for online learning: Importance and confidence.," *Online Lear.*, vol. 24, no. 2, pp. 38–58, 2020.
[48] A. Bifet et al., "Massive online analysis, a framework for stream classification and clustering," in *Proc. 1st Workshop Appl. Pattern Anal.*, 2010, pp. 44–50.
[49] H. M. Gomes et al., "Adaptive random forests for evolving data stream classification," *Mach. Learn.*, vol. 106, pp. 1469–1495, 2017.
[50] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proc. 49th Annu. Meeting Assoc. Comput. linguistics: Hum. Lang. Technol.*, 2011, pp. 142–150.
[51] Y.-N. Zhu and Y.-F. Li, "Semi-supervised streaming learning with emerging new labels," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 7015–7022.

Dianlong You (Member, IEEE) received the PhD degree in computer application technology from Yanshan University, China, in 2014. He is currently the professor and PhD supervisor with the School of Information Science and Engineering, Yanshan University, China. From 2017 to 2018, he was a visiting scholar with the School of Computing and Informatics, University of Louisiana at Lafayette, Lafayette, Louisiana. From 2022 to 2023, he was a visiting scholar with the Data Science Institute, University of Technology Sydney, Sydney, NSW, Australia. His current research interests include machine learning, streaming feature selection and causal discovery. He has more than 20 publications including journals of *IEEE Transactions on Neural Networks and Learning Systems*, *IEEE Transactions on Knowledge and Data Engineering*, *Information Fusion*, *Information Sciences*, and *Knowledge-Based Systems*, etc.

Huigui Yan is currently working toward the PhD degree with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei. His current research interests include streaming feature selection and causal discovery.

Jiawei Xiao is currently working toward the MS degree with the School of Information Science and Engineering, Yanshan University, Qinhuangdao, Hebei. His current research interests include focused on online learning for streaming data and causal discovery.

Zhen Chen received the BS and PhD degrees in computer science and technology from Yanshan University in China, in 2010 and 2017, respectively. He is currently an associate professor. He is currently working on service computing and data mining.

Di Wu (Member, IEEE) received the PhD degree from the Chongqing Institute of Green and Intelligent Technology (CIGIT), Chinese Academy of Sciences (CAS), China, in 2019 and then joined CIGIT, CAS, China. He is currently a professor with the College of Computer and Information Science, Southwest University, Chongqing, China. He has more than 80 publications, including 21 IEEE Transactions papers and several conference papers on AAAI, ICDM, WWW, IJCAI, etc. He is serving as an associate editor for the Neurocomputing and Frontiers in Neurorobotics. His research interests include machine learning and data mining. His homepage: https://wudi1986.github.io/Homepage/.

Limin Shen (Member, IEEE) received the BS degree from Yanshan University, China. He is currently the professor and PhD supervisor with the School of information Science and Engineering, Yanshan University, China. His main research interests include data driven security, service computing, and cooperative defense.

Xindong Wu (Fellow, IEEE) received the bachelor's and master's degrees in computer Sscience from the Hefei University of Technology, China, and the PhD degree in artificial intelligence from the University of Edinburgh, Britain, in 1993. He currently is the director and professor with the Key Laboratory of Knowledge Engineering with Big Data (the Ministry of Education of China), Hefei University of Technology, China. He is a foreign member with the Russian Academy of Engineering, and a fellow of AAAS. He is the Steering Committee Chair of ICDM and the editor in-chief of KAIS. His research interests include Big Data analytics, data mining, and knowledge engineering.